

In the Drawings:

Please replace the original drawing sheet containing FIG. 1 with the attached Replacement Sheet in which the label has been changed to “FIG. 1 (Prior Art)”.

Please replace the original drawing sheet containing FIG. 2 with the attached Replacement Sheet in which the label has been changed to “FIG. 2 (Prior Art)”.

REMARKS

Claims 1 and 10 have been amended to correct typographical errors. Claims 1 – 33 remain pending in the application. Reconsideration is respectfully requested in view of the following remarks.

Drawings:

The Examiner indicated that Figures 1-2 should be designated as Prior Art. Accordingly, replacement drawing sheets in which Figures 1 and 2 are labeled Prior Art are attached herewith.

Section 103(a) Rejections:

The Examiner rejected claims 1-9, 13-24 and 28-33 under 35 U.S.C. § 103(a) as being unpatentable over APA in view of Jin et al. (U.S. Patent 6,330,689) (hereinafter “Jin”), and claims 10-12 and 25-27 as being unpatentable over APA in view of Jin, and further in view of Crites et al. (U.S. Patent 6,097,380) (hereinafter “Crites”).

Claim 1 recites a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process node. In rejecting claim 1, the Examiner acknowledges that APA (the background section of the present application) “does not explicitly teach wherein the plurality of nodes in the system comprises at least one in-process node and at least one [out-of-]process node”. In fact, the background section clearly and unequivocally teaches that “Conventional systems allow *only one type of configuration* - either every node is an in-process node or every node is an out-of-process node. For example, if an out-of-process client is desired, then all other clients would also need to be configured as out-of-process clients” (Page 4, lines 20 – 23). See also, with respect to Figure 1: “This configuration is referred to as out-of-process and all nodes in the distributed data system in this configuration are configured out-of-process”(Page 1, line 31 – page 2, line 3) and with respect to Figure 2: “This

configuration is called in-process and all nodes in the distributed data system in this configuration are configured in-process” (Page 3, lines 16 – 18). Thus, the APA reference relied upon by the Examiner in rejecting claim 1 explicitly teaches away from the combination of limitations recited in claim 1. “References that teach away cannot serve to create a prima facie case of obviousness.” *In re Gurley*, 27 F.3d 551, 553, 31 USPQ2d 1131, 1132 (Fed. Cir. 1994).

Further with respect to claim 1, the Examiner asserts that “Jin teaches a data system comprising wherein the plurality of nodes in the system comprises at least one in-process node and at least one process node (col. 6 lines 42 – 48).” The Examiner then further asserts that “it would have been obvious to one of ordinary skill in the art to modify APA to include both in-process and out-of-process nodes as in Jin. One would be motivated to do so to offer the flexibility to run either or both in-process and out-of-process applications (Jin, col. 6 lines 56 – 57).” Applicants strongly disagree with the Examiner’s assertions for a number of reasons.

First, contrary to the Examiner’s suggestion, Jin does not teach or suggest an in-process node of a distributed data system, where the in-process node comprises a client and a distributed data manager configured to execute within the same process, as recited in claim 1. In contrast, Jin teaches “application managers” that may run “within a server’s process (i.e., in-process)” or “within separate processes (i.e., out-of-process)”. The cited lines of Jin are reproduced below:

The application managers 64 may run within the server's process (i.e., in-process), or within its own separate process (i.e., out-of-process) either on the same machine as the server or on a different machine. In FIG. 4, application managers 64(1) and 64(2) are in-process with the server program 62, whereas application manager 64(3) is out-of-process with the server program 62. In-process applications allow maximum efficiency at the risk of bringing down the server in the event that the application is not stable. Alternatively, out-of-process applications are effectively isolated so that if an application fails, it should not affect the running of the server. However, this isolation benefit comes at the cost of lower performance when compared to an in-process application. The server architecture 60 advantageously offers the developers the flexibility to run either or both in-process and out-of-process applications. (Jin, col. 6, lines 42 – 57)

Jin teaches that an “application manager” is a “processor of client requests”, and is thus clearly very different from a “client”: see, e.g., col. 6, lines 20 – 29:

Conceptually, the application manager 64 is a dynamic processor of client requests that uses one or more agents to generate content to be returned to the client. The application manager 64 dynamically loads associated agents 66 as needed to handle particular user requests. The application manager 64 can handle an arbitrary number of user requests at a time, and it can load and maintain an arbitrary number of agents 66 into its address space in order to process user requests.

Thus Jin’s use of the terms “in-process” and “out-of-process” refers to the manner in which Jin’s “application managers” responsible for processing client requests interact with Jin’s “server processes”. Jin does not teach or suggest a client executing within the same process as a distributed data manager, much less an in-process node of a distributed data system at which a distributed data manager and a client execute within the same process, as recited in claim 1. Jin is also silent with respect to the use of “non-serialized format” and “serialized format” for communication with different entities by a distributed data manager of an in-process node, as also recited in claim 1. There is no teaching or suggestion in the art of record that Jin’s in-process and out-of-process application manager and server combinations would in anyway apply to the configuration of clients and distributed data managers in the nodes of APA.

Further with respect to claim 1, even if the teachings of Jin were to be combined with those of APA, this would only lead to “application managers” (client request processors) that may “run within the same process” or “different processes” as a server, not to a distributed data system in which at some nodes, clients run in the same process as distributed data managers, and at other nodes, clients run in separate processes from distributed data managers, as recited in claim 1.

Moreover, APA expressly teaches away from the combination proposed by the Examiner. As noted above, APA teaches that in conventional distributed data systems, all nodes must be either in-process or out-of-process (no mixing of in-process and out-of-process nodes is allowed). It is improper to combine references where the references

teach away from their combination. *In re Grasselli*, 218 USPQ 769, 779 (Fed. Cir. 1983). Therefore, the rejection is clearly improper.

For at least the reasons cited above, the rejection of claim 1 is not supported by the cited art.

With respect to claim independent claims 17 and 32, in each case the Examiner once again acknowledges that APA “does not explicitly teach in-process node and out-of-process node are in the distributed data system” and asserts that the limitations of the claims are rendered obvious by the combination of APA with the same portions of Jin that were cited in the rejection of claim 1. The Examiner’s assertions with respect to claim 1 have already been shown to be in error. Accordingly, the rejection of claims 17 and 32 is also unsupported by the cited art.

With respect to claim 6, the Examiner alleges that the limitation “wherein all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node” is taught in APA page 4, lines 15 – 18. The Examiner is incorrect. Neither the cited lines nor any other portion of APA teaches **all** data store operations result in data being stored in non-serialized object format by a distributed data manager of an in-process node. Similar remarks apply in regard to claim 12.

With respect to claim 10, the Examiner asserts that Crites teaches “the distributed data manager for the out-of-process node is configured to store the data in its serialized form”, citing col. 2, lines 36 – 42 and col. 5, lines 19 – 20 of Crites in support. The Examiner is incorrect in this interpretation of Crites. The cited lines of column 2 of Crites merely state that “the continuous media streams consist of sequences of digital data that are intended to be supplied serially to client computers and rendered by the client computers in a form that is useful to users of the client computers”. Crites provides an example of a continuous media stream as “a sequence of audio samples” that are sent to a “client computer” for “rendering as audible sound”. In contrast, claim 10 recites the “out-

of-process client” sending “serialized data” to the distributed data manager at an out-of-process node of a distributed data system, and the distributed data manager then storing the data in serialized format. In Crites, the “media stream” is sent to the client, whereas in claim 10, serialized data is sent from the client to a distributed data manager for storage. Furthermore, data that is merely “supplied serially” as taught by Crites is not the same as data sent in “serialized format”, as recited in claim 10. Column 5, lines 19 – 20 of Crites merely teach that “the services or functions of an object” (i.e., a programming object representing a media stream, see col. 5, lines 1 – 3) “can be used by both in-process clients and by out-of-process clients”, which appears to have nothing to do with the limitations recited in claim 10.

Moreover, even if the teachings of Crites were to be combined with those of Jin and APA, the resulting system would merely allow media streams to be served from the nodes of a homogeneous distributed data system (i.e., one in which the nodes are either all “in-process nodes” or all “out-of-process nodes”), and would not render the combination of limitations of claim 10 obvious. Thus, the rejection of claim 10 is clearly unsupported by the cited art.

The Examiner asserts that claim 11 “does not teach or define any new limitation above claim 10”. The Examiner is incorrect. Claim 11 recites the limitations “wherein the distributed data manager for the out-of-process node is configured to receive serialized data from another node” and “store the data in its serialized format”, which are not recited in claim 10 or its parent claims. Thus, the Examiner has failed to state a *prima facie* rejection of claim 11.

Applicants also assert that numerous ones of the remaining dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

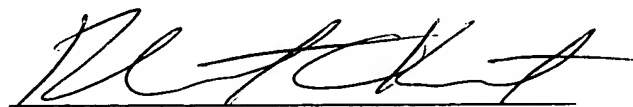
Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-05700/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☒ Two Replacement Drawing Sheets (FIGs. 1 & 2)

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: December 19, 2005